

- Building our own Create React App.
- JS Engine
- Injecting React in an APP.
- Creating an Element in React
- Props in React
- Production Ready APP.
  - minified
  - bundle
  - server
  - optimization
  - clean console
  - caching
  - treeshaking
- BUNDLERS
  - webpack
  - parcel
  - vite

It is a measure of packages (js files)
- In create React App use is webpack
  - the renderers they
  - using package manager.
  - install package
  - NPM or Yarn
- what is NPM?
  - NPM doesn't stand for Node Package Manager.
  - there is no any official name of NPM.

→ How to get npm in an app?  
npm init →  
↓ creates  
package.json > This is the configuration  
which npm needs.

→ npm basically manages all as packages.  
Q Why do we use npm?  
Because while making projects, we  
need lot of packages to make an  
app work which can't be done by  
single tech stack.

→ Lot of helper packages are managed  
by npm.

Igniting the App → to  
• npm install [package name] → to  
install any package in npm.

→ Installing parcel

npm install [-D] {parcel  
[-cain-dev]}

their dependency

all the packages  
which our project  
needs.

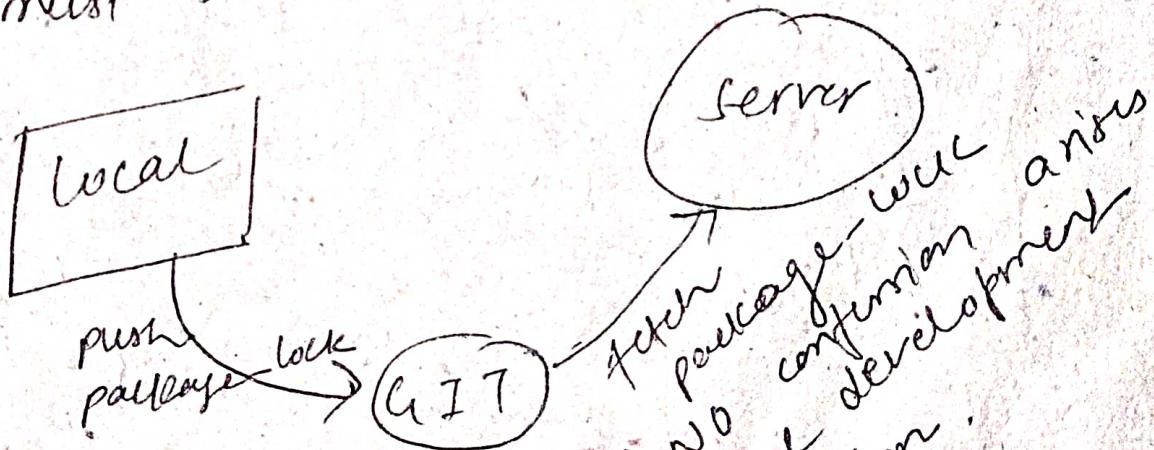
package-lock.json created.  
node modules comes up.

Dev dependency means we need that package in our development environment.

- caret and tilde in package.json  
(`~`)
- package-lock gives the exact version as system have.
- It locks the version which we use.

The famous confusion that my app is work on my local machine but not production comes from here only if our package version mismatches.

# Never put package-lock in gitignore  
It must be in our repository.



→ It maintains the integrity the version of package is same on local as well as and production.

node-modules >  
→ whenever we install something (package)  
it gets stored in node-modules only.

→ If we use as a database for npm.  
Q Should we add node-modules to

gitignore? It is foolish to put it

Ans Yes, it is foolish to put it  
in a repository. (It's heavier).

It doesn't make sense to put it  
on github bcz our package-lock.json  
have sufficient information to  
create node-modules (or generate)  
it on server.

→ CDN is not a good way to  
inject React in an App.

↳ Good way to use packages

i) why? Because in future if version gets  
updated we have to manually update  
the scripts tag (NOT viable).

ii) CDN links are not stored on our server.

It is faster to fetch from our server  
rather than some remote server.

iii) what if CDN link gets down.

→ Installing React as package in  
Igniting our App

npm install react

Installing react-dom

npm install react-dom  
(i)

→ Igniting the App

use command > index.html

npx parcel

↓

execute using  
npm

entry point

from we want  
to execute

It will create a local server.

→ Importing React  
import React from "react";  
import it from node-modules,  
import react-dom from "react-dom/client";

import react-dom from node-modules,

# Never ever edit node-modules/  
package.json

modules can be imported or exported

scripts can't

<script type="module" src="App.js"></script>

<script type="

- Live Server functionality implements by parcel using hot module replacement.  
it tracks each and every DOM object whenever something changes it reflects the same on screen as well.

→ Parcel does:

- HMR
- file watcher algorithm (written in C++)  
It uses parcel-cache folder to do all these steps:  
→ parcel will push all the build file in dist folder. (production build)  
contain the minified version.
- Bundling
- minify
- console log
- clean maximum time to load on the websites (image optimization)
- dev about product build algorithm
- super fast build algorithm
- caching while development
- compression of file

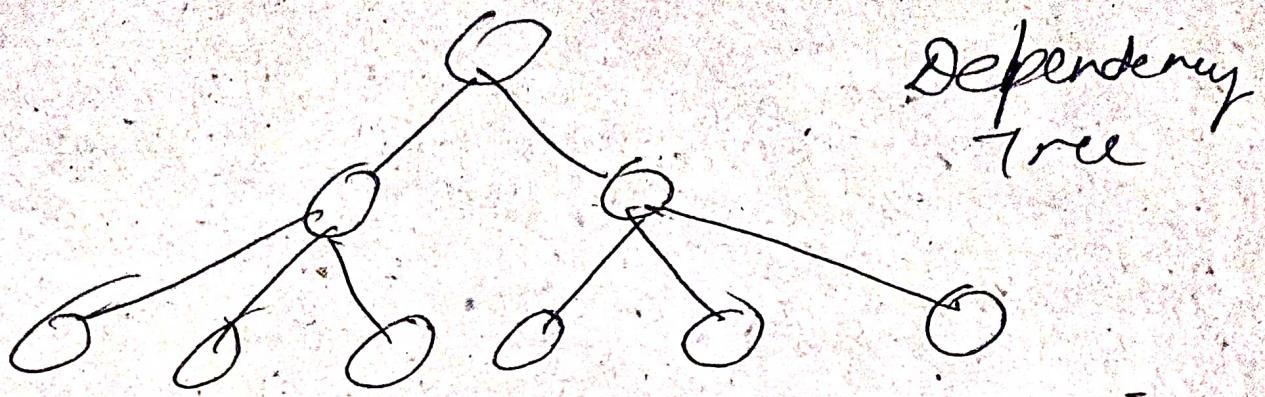
- make an app compatible across older version of browsers  
(add polyfills).
- give the functionality of enabling HTTPS on dev testing [useful while number manages the port number]
- manages the port - cache in # we should put parcel - cache in .gitignore
  - ↳ Anything which can be generated on our server will be put in .gitignore.

uses Consistent Hashing Algorithm  
to do the bundling  
why React is performant or fast?

Parcel is zero config bundler.

# Transitive Dependency

→ package manager manages transitive dependencies (one package depends on others and in turn it depends on some other package).



→ Use case of browserlist: [

    → config

    ] put it ~~in~~ in package.json  
to make an app compatible for  
older or specific browsers  
well.